CAKE: Cloud Architecture Knowledge Evaluation of Large Language Models

Tim Lukas Adam
Centre for Industrial Software
University of Southern Denmark
Alsion 2, Sønderborg, 6400, Denmark
tiada23@student.sdu.dk

Phongsakon Mark Konrad
Centre for Industrial Software
University of Southern Denmark
Alsion 2, Sønderborg, 6400, Denmark
phkon23@student.sdu.dk

Riccardo Terrenzi
Centre for Industrial Software
University of Southern Denmark
Alsion 2, Sønderborg, 6400, Denmark
rite@mmmi.sdu.dk

Florian Girardo Lukas
Centre for Industrial Software
University of Southern Denmark
Alsion 2, Sønderborg, 6400, Denmark
fllu@mmmi.sdu.dk

Rahime Yilmaz
Centre for Industrial Software
University of Southern Denmark
Alsion 2, Sønderborg, 6400, Denmark
rayil@mmmi.sdu.dk

Krzysztof Sierszecki
Centre for Industrial Software
University of Southern Denmark
Alsion 2, Sønderborg, 6400, Denmark
krzys@mmmi.sdu.dk

Serkan Ayvaz
Centre for Industrial Software
University of Southern Denmark
Alsion 2, Sønderborg, 6400, Denmark
seay@mmmi.sdu.dk

***Abstract*—In today's software architecture, large language models (LLMs) serve as software architecture co-pilots. However, no benchmark currently exists to evaluate large language models' actual understanding of cloud-native software architecture. For this reason we present a benchmark called CAKE, which consists of 188 expert-validated questions covering four cognitive levels of Bloom's revised taxonomy—recall, analyze, design, and implement—and five cloud-native topics. Evaluation is conducted on 22 model configurations (0.5B–70B parameters) across four LLM families, using three-run majority voting for multiple-choice questions (MCQs) and LLM-as-a-judge scoring for free-responses (FR). Based on this evaluation, four notable findings were identified. First, MCQ accuracy plateaus above 3B parameters, with the best model reaching 99.2%. Second, free-response scores scale steadily across all cognitive levels. Third, the two formats capture different facets of knowledge, as the MCQ accuracy approaches a ceiling while free-responses continue to differentiate models. Finally, reasoning augmentation (+think) improves free-response quality, while tool augmentation (+tool) degrades performance for small models. These results suggest that the evaluation format fundamentally shapes how we measure architectural knowledge in LLMs.**

***Index Terms*—software architecture, large language models, benchmark, cloud-native, architectural knowledge, Bloom's taxonomy**

I. Introduction

Recently, Large language models (LLMs) are becoming co-pilots for software engineering, from writing code to making architectural decisions [12]. Cloud-native software architecture covers microservices, containerization, orchestration, and cloud deployment patterns. In this domain, architectural choices have consequences for scalability, resilience, and maintainability [9], [10].

Today's benchmarks test LLMs on code generation (SWE-bench [1], HumanEval [2]), code reasoning (CRUXEval [4]), or broad knowledge (MMLU [5], BIG-bench [6]). ArchCode [3] examines architecture-related code understanding but remains at the code level rather than spanning cognitive levels of architectural knowledge. No current benchmark evaluates whether LLMs grasp the conceptual and procedural knowledge behind cloud-native architecture decisions.

Software architects increasingly use LLMs throughout the software architecture lifecycle, e.g., to gather and refine requirements, explore design alternatives and trade-offs, and draft architectural documentation and decisions [12]. Therefore, practitioners need to understand where LLM assistance is reliable and where human judgment remains essential. With the goal of establishing a benchmark for cloud-native software engineering tasks, we address this gap with three contributions:

1) CAKE, a benchmark for cloud-native software architecture knowledge across four cognitive levels of Bloom's revised taxonomy [13], containing 188 expert-validated questions across five cloud-native topics.
2) Empirical findings from testing 22 model configurations (0.5B–70B parameters) across four LLM families, including base, reasoning-enhanced (+think), and tool-augmented (+tool) variants.
3) Public artifacts, including the benchmark dataset[1].

The remainder of this paper is organized as follows. Section II reviews related work and Section III introduces the benchmark CAKE. Section IV reports results, Section V

[1]https://github.com/timadam03/CAKE-benchmark

discusses implications and limitations, and Section VI presents the conclusions.

II. Related Work

Recent LLM benchmarks focus on code generation and reasoning. SWE-bench [1] tests models on real-world GitHub issues, HumanEval [2] targets function-level code generation, CRUXEval [4] measures code reasoning, and ArchCode [3] introduces architecture-aware code generation tasks. All remain at the code level, none testing the architectural knowledge behind design decisions. Broad knowledge benchmarks like MMLU [5], MMLU-Pro [25], and BIG-bench [6] cover many domains but neglect software architecture. GPQA [7] covers graduate-level science but not software engineering. QuArch [19] benchmarks LLMs on computer architecture — a field distinct from software architecture — nevertheless its use of Bloom's cognitive levels inspired this benchmark's design.

Domain-specific benchmarks have adopted cognitive-level frameworks. LawBench [21] applies Bloom's taxonomy to legal reasoning. Automated Bloom's-aligned generation [22] demonstrates that such stratification generalizes across domains. In engineering, DesignQA [23] tests engineering documentation comprehension. A recent systematic review of software architecture and LLMs [24] identifies evaluation of architectural knowledge as a key open challenge.

Concept inventories [11] and Bloom's taxonomy [14], [15] provide frameworks for multi-level knowledge testing. Bass et al. [9] and Richards and Ford [10] shaped our question design. On the evaluation side, LLM-as-a-judge [8] enables scalable open-ended scoring but raises reliability concerns including language-dependent biases [26] and prompt sensitivity [20]; we mitigate these with a single deterministic judge and rubric-based scoring. Our pipeline combines three-run majority voting for multiple-choice questions(MCQ) with LLM-as-a-judge for free-response (FR).

III. The CAKE Benchmark

A. Design

CAKE targets cloud-native software architecture: microservices, containers, orchestration, and cloud-managed services. We drew 85 core concepts from several sources, including the AWS Well-Architected Framework, microservices.io, Azure Architecture Patterns, and Kubernetes Enhancement Proposals.

Questions cover four cognitive levels mapped to Bloom's revised taxonomy [13]: recall, analyze, design, and implement, spread across five topics: architectural patterns, quality attributes, decomposition strategies, cloud deployment, and technical debt. The benchmark holds 130 MCQ and 58 free-response questions (188 total): recall 50, analyze 60, design 50, implement 28. Implementation knowledge is assessed exclusively through free-response. The question distribution is illustrated in Fig. 1.

B. Generation and Validation

Claude Opus 4.5 generated all questions using a parametric prompt template that specified the target topic, skill level, difficulty tier, and response format. Questions were generated in batches per topic–skill cell, with manual review between batches to check diversity.

Four domain experts independently rated the 200 candidate questions on clarity (1–5), correctness (1–5), and difficulty accuracy (1–5), also flagging ambiguity or typos. Three experts completed all annotations, while one provided partial ratings. In total, the 658 annotations indicate high quality (Fig. 2): mean clarity 4.72 out of max. 5, mean correctness 4.63 out of max. 5. Pairwise agreement within one scale point reaches 91.3% for clarity and 77.4% for correctness. Ordinal Krippendorff's α [16] is near zero for both ($\alpha = 0.00$ and $\alpha = -0.01$), reflecting a ceiling effect rather than genuine disagreement: when most ratings fall between 4 and 5, the remaining variance becomes difficult to interpret. After validation, we excluded 12 implementation-level MCQs due to a formatting defect in the options array (Section V-C). Because implementation knowledge is better captured through free-response questions, 188 questions remained for evaluation.

TABLE I
Evaluated Model Families and Configurations.

Family	Sizes	Base	+think	+tool	Total
Qwen	0.5–30B	6	–	–	6
Llama	1–70B	4	3	3	10
Mistral	3–14B	3	1	–	4
GPT	4o-Mini, 5-Mini	2	–	–	2
Total		**15**	**4**	**3**	**22**

Count of configurations per family. Configurations consist of base (direct question answering), +think (structured reasoning with Chain-of-Thought), and +tool (agentic tool use enabled).

C. Models

We tested 22 model configurations from four families spanning 0.5B to 70B parameters (Table I). Locally-served models from the Qwen, Llama and Mistral families run on Ollama[2] and GPT models run on OpenRouter[3]. All use temperature 0 to reduce randomness and create more consistent outputs across runs. Three execution modes are available per model, the baseline (direct question-answer), structured reasoning (+think, turning on chain-of-thought by prepending a reasoning preamble and raising the maximum token limit from 256 to 2048), and agentic tool use (+tool, which supplies web search and browsing functions). These modes map to the config column in Table I. The reports of the full per-configuration results are presented in Table II.

D. Evaluation Pipeline

The evaluation pipeline follows a provider-agnostic design supporting three inference backends: Ollama (CLI runtime), LM Studio[4] (GUI runtime), and OpenRouter (cloud API). Each provider module exposes a shared interface for model

[2]https://ollama.com
[3]https://openrouter.ai
[4]https://lmstudio.ai

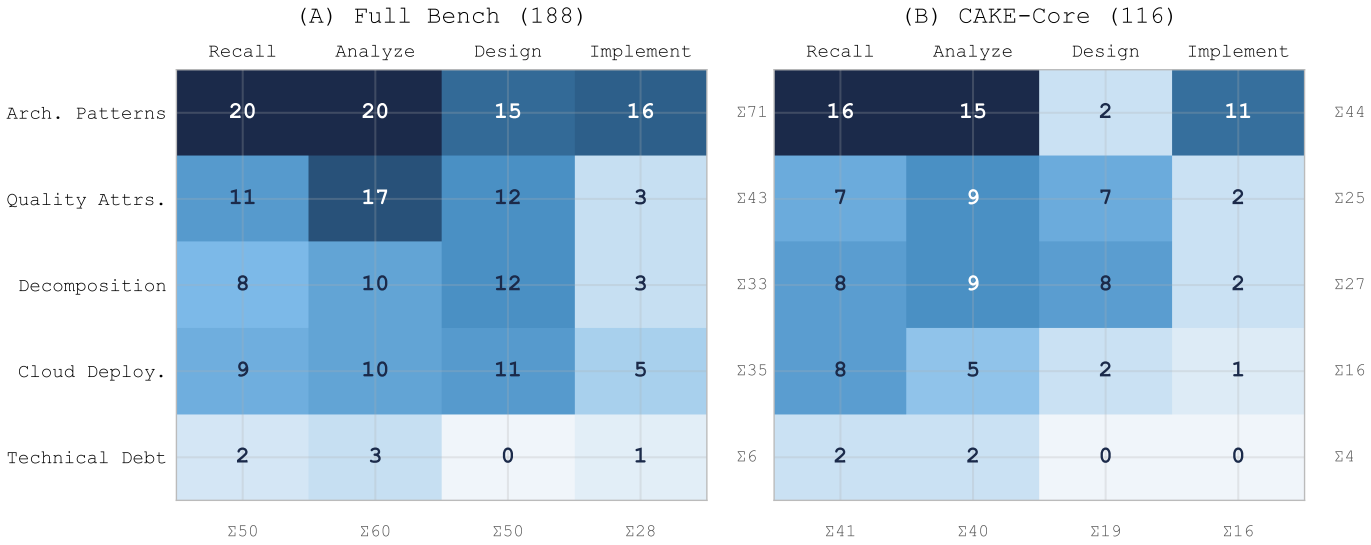

Fig. 1. Question distribution across five cloud-native topics and four cognitive levels. (A) Full Bench (188 evaluated questions; 12 implement-level MCQs excluded due to a formatting defect). (B) CAKE-Core subset (116 questions passing mean correctness ≥ 4.0 and clarity ≥ 4 filters, with no flags). Cell values show question counts; color intensity indicates density.

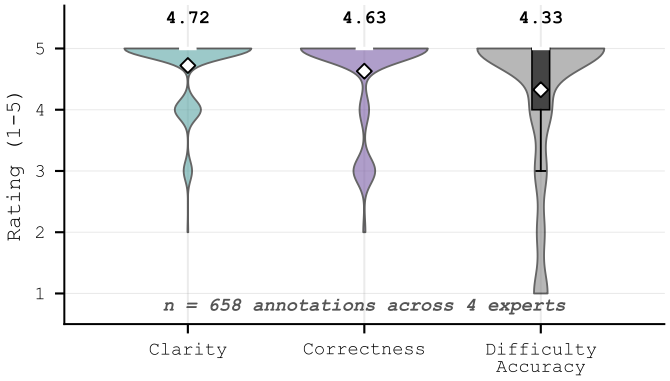

Fig. 2. Distribution of expert ratings across clarity, correctness, and difficulty accuracy. Violin plots show density; box overlays show median and IQR. High means (4.72, 4.63) with narrow spread confirm benchmark quality.

initialization and response generation, so models can be replaced without adjusting the evaluation logic.

The diverse output formats are extracted through a multi-stage priority chain, including explicit pattern matching, stripping of thinking tags for reasoning-mode outputs, JSON parsing, and regex matching. This layered approach is necessary because different models and configurations produce answers in varying formats.

E. MCQ Evaluation

Each MCQ is presented three times with shuffled options, with a majority voting reducing positional bias and capturing answer consistency. The score is the fraction of runs that agree with the majority answer: 1.0 (unanimous) signals high confidence, while 2/3 (split majority) flags uncertainty. Across all configurations, unanimous answers reach 89.5% accuracy versus 55.0% for split-majority answers, a 34.5 percentage-point gap confirming conviction as a meaningful confidence signal.

F. Free-Response Evaluation

Free-response questions are graded using LLM-as-a-judge [8], with DeepSeek-R1:32B as the judge model on a deterministic 0–5 grading scale. The judge receives the question, reference rubric, and model response, then produces a single numerical score. DeepSeek-R1 was chosen for its consistency and stable scoring behavior across multiple runs.

IV. Results

Our study results are presented in Table II for all 22 configurations, grouped by model family. The results are organized into four categories: MCQ performance, free-response evaluation, augmentation effects, and overall trends across formats.

A. MCQ Performance

MCQ accuracy approaches a ceiling for most configurations. Sixteen of 22 models exceed 90%, and within each family the scaling of the base model is monotonic. Above 3B parameters, gains shrink to 1–3 percentage points per size step. GPT models achieve perfect or near-perfect MCQ accuracy. Across local models, performance converges rapidly once models exceed the 3B threshold. Resource usage varies slightly, with inference time ranging from 0.9 s for Qwen 3B to 8.8 s for Llama3.1 8B +tool per question. Augmented variants do not significantly increase VRAM usage, but introduce longer inference times, with +think increasing latency by approximately 3–7×.

TABLE II
PERFORMANCE AND RESOURCES FOR ALL 22 CONFIGURATIONS, GROUPED BY FAMILY (↑ HIGHER IS BETTER). GPT MODELS SHOW "—" FOR PARAMS AND VRAM AS THEY RUN VIA OPENROUTER.

Model	Params	Cfg	MCQ (%)↑	FR Overall↑	Time (s)↓	VRAM (GB)
Qwen						
Qwen 2.5	0.5B	base	47.7	1.99	1.2	30.5
Qwen 2.5	1.5B	base	91.5	2.71	1.0	31.8
Qwen 2.5	3B	base	95.4	3.06	**0.9**	28.7
Qwen 2.5	7B	base	96.9	3.02	1.1	28.7
Qwen 2.5	14B	base	97.7	3.36	1.5	28.7
Qwen3-Coder	30B	base	**99.2**	4.04	1.8	29.0
Llama						
Llama 3.2	1B	base	37.7	2.38	1.0	28.7
Llama 3.2	1B	+think	76.2	2.76	4.4	24.2
Llama 3.2	1B	+tool	13.8	1.71	2.5	24.2
Llama 3.2	3B	base	93.8	2.86	1.3	28.7
Llama 3.2	3B	+think	93.8	3.01	4.9	25.4
Llama 3.2	3B	+tool	40.8	1.75	2.5	25.4
Llama 3	8B	base	93.8	2.96	1.2	28.7
Llama 3	8B	+think	95.4	2.94	6.3	28.7
Llama 3.1	8B	+tool	91.5	2.85	8.8	28.1
Llama 3.3	70B	base	**99.2**	3.47	3.7	—
Mistral						
Mistral 3.1	3B	base	96.9	3.54	1.3	29.0
Mistral 3.1	3B	+think	83.8	3.96	8.6	26.3
Mistral 3.1	8B	base	98.5	4.08	1.7	29.0
Mistral 3.1	14B	base	**99.2**	**4.33**	2.4	29.0
GPT						
GPT-4o-Mini	—	base	**99.2**	3.77	1.9	—
GPT-5-Mini	—	base	**99.2**	**4.52**	10.0	—

B. Free-Response Analysis

Free-response scores span a much wider range than MCQs, from 1.71 for Llama3.2 1B +tool to 4.52 for GPT-5-Mini. This spread exposes capability gaps that MCQ results do not capture. Implement-level free-response is the primary differentiator. The score range at this level, 1.36–4.54, exceeds the spread at analyze, 2.00–4.75, and design, 1.31–4.31. Implementation is evaluated exclusively through free-response, reinforcing the value of generative evaluation for higher-order architectural tasks.

Fig. 3 ranks all configurations by judge score across cognitive levels. Mistral 14B and GPT-5-Mini consistently occupy the top two positions. Rankings vary across panels, indicating that cognitive-level scores are not perfectly correlated. GPT-4o-Mini places 6th on analyze but 5th on implement, while Llama3.3 70B ranks higher on implement than on design. The gap between MCQ and free-response performance is most pronounced for mid-size models in the 3–8B range, which often select correct MCQ answers but struggle to articulate architectural reasoning.

Within each family, free-response performance increases with parameter count, though gains diminish at mid-range sizes. In the Qwen family, FR overall rises from 1.99 at 0.5B to 3.06 at 3B and 4.04 at 30B, with a slight decline between 3B and 7B (−0.04). Llama shows a similar plateau, increasing from 2.38 at 1B to 2.86 at 3B and 2.96 at 8B. Mistral displays steady improvements at every size tier, from 3.54 at 3B to 4.08 at 8B and 4.33 at 14B. This family-specific scaling pattern suggests that architectural knowledge gains depend on training data composition [18] rather than parameter count alone.

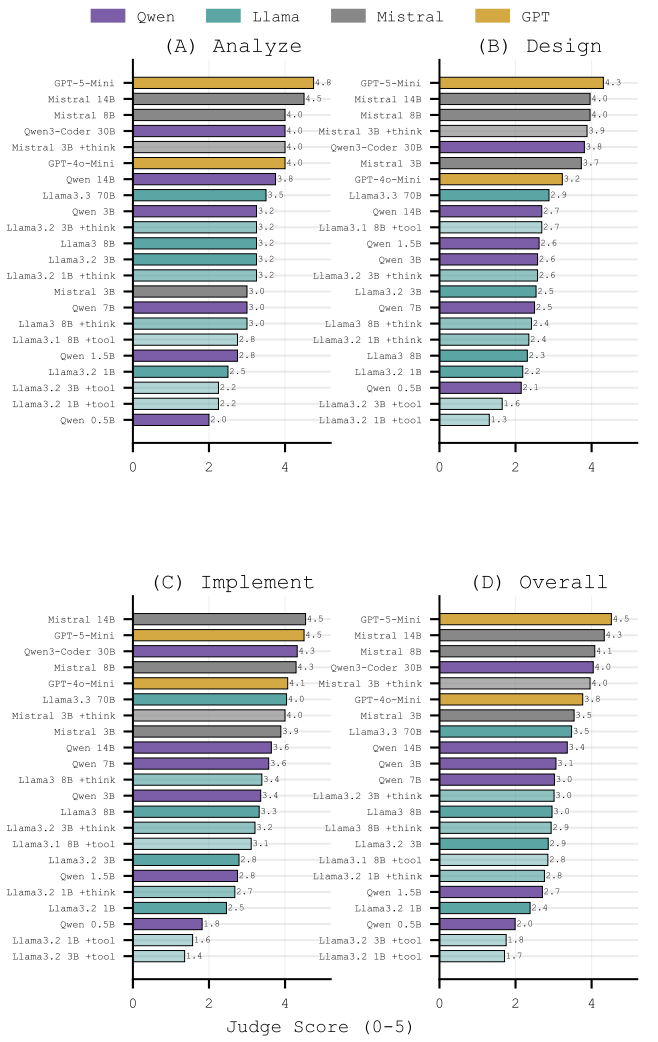

Fig. 3. Free-response judge scores (0–5) for all 22 configurations, ranked per panel, (A) Analyze, (B) Design, (C) Implement and (D) Overall.
Each panel shows one cognitive level; color indicates model family. Mistral models consistently rank highest among local models.

C. Augmentation Effects

MCQ and FR deltas for the four `+think` and three `+tool` variants against their base models are plotted in Fig. 4.

`+think` generally improves free-response quality (+0.15 to +0.42 FR overall), with only Llama3 8B showing a marginal decline (−0.02), while MCQ effects are mixed. Llama3.2 1B gains +38.5 pp while Mistral 3B drops −13.1 pp. A size-dependent trade-off can be observed. For Llama3.2 1B, reasoning augmentation nearly doubles MCQ accuracy (37.7%→76.2%) as well as lifting free-responses (+0.38), suggesting that chain-of-thought can compensate for limited parametric knowledge. At 3B, the MCQ benefit disappears (93.8%→93.8%) while free-response still gains (+0.15).

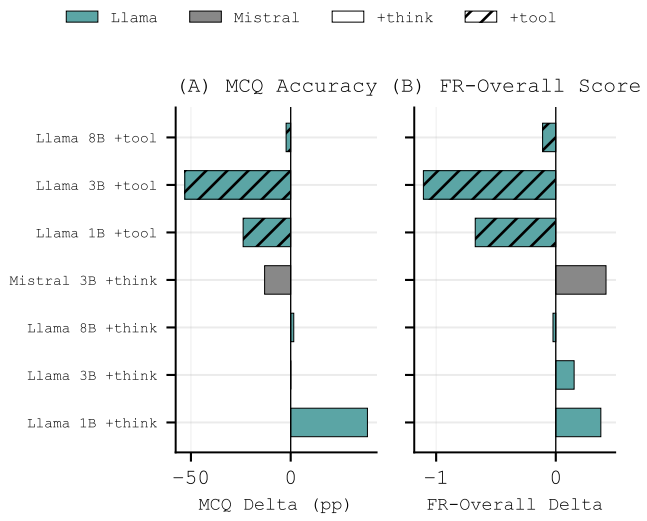

Fig. 4. Augmentation effects on MCQ and Free-response performance. (A) MCQ accuracy delta. (B) FR overall delta for +think and +tool variants relative to their base models. Color indicates model family, hatching indicates +tool.

For Mistral 3B, +think degrades MCQ results by 13.1 pp despite producing the largest free-response gain (+0.42).

`+tool` negatively impacts performance for small models. The degradation follows a clear size gradient: Llama3.2 1B drops −23.9 pp MCQ and −0.67 FR overall, Llama3.2 3B drops −53.0 pp and −1.11, while Llama3.1 8B loses only −2.3 pp MCQ and −0.11 FR overall. Only the 8B variant keeps reasonable performance (91.5% MCQ, 2.85 FR overall), pointing to a minimum capacity threshold around 8B parameters for effective tool use.

D. Overall Performance

Across formats, GPT-5-Mini achieves the highest free-response score at 4.52, while Mistral 14B leads among local models at 4.33. MCQ performance converges rapidly above 3B parameters, whereas free-response continues to differentiate models across the full size range.

Mistral achieves the strongest free-response results at each local size, with Mistral 3B outperforming larger Qwen 7B and Llama3 8B models. GPT models achieve near-ceiling MCQ accuracy yet diverge substantially on free-response, indicating that multiple-choice performance does not reliably reflect generative architectural competence.

V. Discussion

A. Scaling and Format Effects

This study provides a benchmark for cloud-native software architecture knowledge, addressing the current lack of a dedicated evaluation resource in this domain. The results indicate that generative evaluation provides a more informative view of architectural understanding than multiple-choice evaluation alone. Implement-level scores capture procedural knowledge with Mistral 14B reaching 4.54 out of max. 5 at the implement level and GPT-5-Mini reaching 4.50 out of max. 5. Training data composition continues to matter. This can be seen with Mistral 3B outperforming Qwen 7B on free-response questions despite having fewer parameters.

At the topic level (Fig. 7), the models show systematic strengths and weaknesses that fade with scale. On recall-level MCQ, Qwen 0.5B scores 72.7% on quality attributes but only 35.0% on architectural patterns and 0% on technical debt. This gap suggests that sub-billion-parameter models retain basic quality vocabulary (e.g., availability, scalability) yet lack coherent knowledge of complex patterns such as CQRS, saga, or sidecar. Above 3B parameters all five topics converge to $\geq$90%, and this convergence holds at the design level too. One exception stands out, which is cloud deployment at the design level, where Qwen 0.5B scores 0% but every 3B+ model hits 100%, pointing to a sharp capability threshold for deployment-related reasoning.

Conviction rates mirror this size dependency. At recall level, Qwen 0.5B gives unanimous answers on just 56% of questions (28/50), while Qwen 7B reaches 96% (48/50) and GPT-5-Mini hits 100%. This rapid jump shows that answer stability emerges alongside accuracy gains, reinforcing conviction as a useful indicator of model confidence rather than mere response consistency.

The conviction metric also offers a practical confidence signal for practitioners. Unanimous answers (conviction = 1.0) hit 89.5% accuracy, while split-majority answers (conviction = 2/3) fall to 55.0%. That 34.5 percentage-point gap suggests conviction could work as a filter in production: an LLM assistant could flag low-conviction architectural suggestions for human review.

To check how question quality affects results, we built CAKE-Core: a subset of the 188 evaluated questions where every expert rated clarity $\geq$ 4, mean correctness $\geq$ 4.0, and no flags. Of these, 116 pass (94 MCQ, 22 free-response), distributed as 41 recall, 40 analyze, 19 design, and 16 implement. Fig. 5 compares Full Bench and CAKE-Core MCQ accuracy across the 94 core MCQ items for all 22 configurations. Accuracy differences are negligible: Qwen 0.5B shifts from 47.7% to 47.9% (+0.2 pp), Qwen 14B from 97.7% to 97.9% (+0.2 pp), and GPT-5-Mini from 99.2% to 98.9% (−0.3 pp). The near-zero deltas indicate that the quality filter neither inflates nor deflates scores, confirming that the full benchmark is well-calibrated and that CAKE-Core preserves the original ranking order.

We also looked at expert–model alignment (Fig. 8). Panel (A) shows that per-question expert difficulty ratings don't predict model accuracy (Spearman $r_s = -0.11$, $p = 0.208$). Human and model difficulty perceptions diverge: questions experts call hard are not systematically harder for LLMs, and vice versa. Panel (B) asks whether expert quality flags (ambiguity or typo) affect model performance. Flagged questions ($n = 15$) and unflagged questions ($n = 115$) yield nearly identical mean model accuracy (83.9% vs. 83.7%), suggesting that flagged questions aren't inherently harder for models but instead reflect surface-level issues human annotators notice

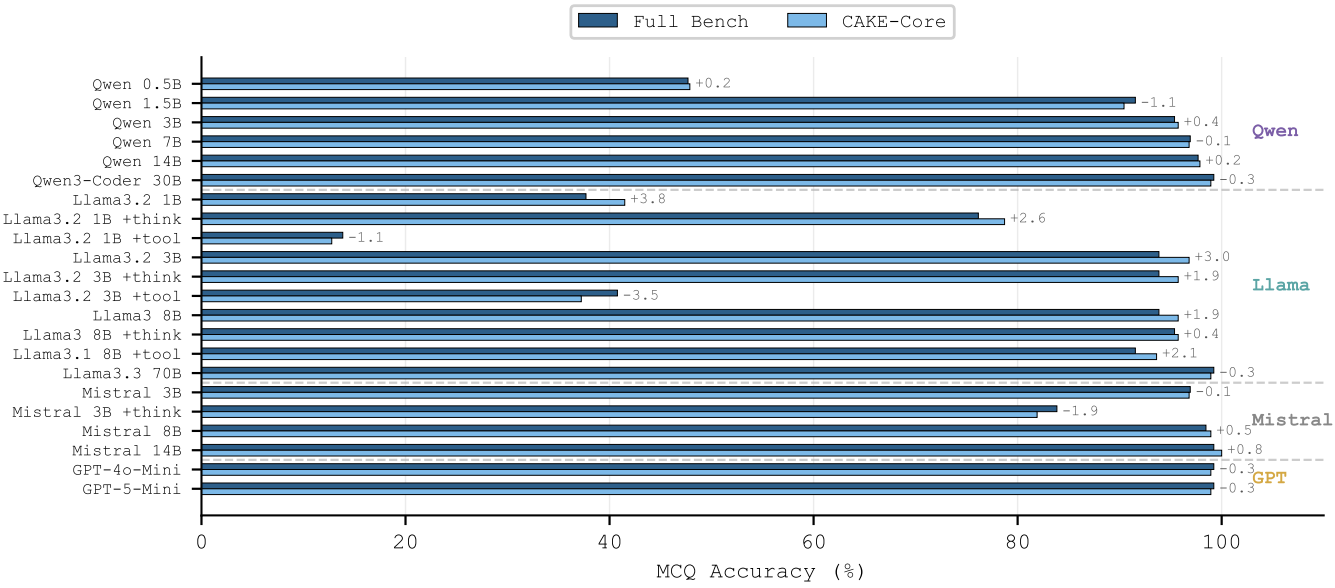

Fig. 5. Full Bench vs. CAKE-Core MCQ accuracy for all 22 configurations. Darker bars show Full Bench; lighter bars show CAKE-Core. Delta values at bar ends indicate the accuracy change after quality filtering. Ranking order is largely preserved across model families.

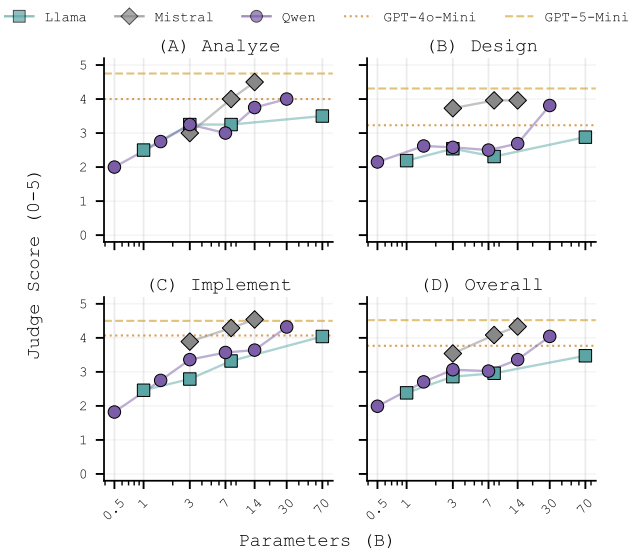

Fig. 6. Free-response judge scores (0–5) vs. model parameters across cognitive levels. (A) Analyze, (B) Design, (C) Implement, and (D) Overall. Scores scale consistently with size for all families; GPT model baselines shown as dashed lines. Unlike MCQ (which saturates above 3B), free-response continues differentiating across the full parameter range.

yet LLMs bypass through pattern-matching. Even so, expert correctness ratings remain a meaningful quality signal, since questions unanimously rated 5.0 for correctness produce the highest mean model accuracy (85.3%) and conviction (75.8%), compared to 67.0% accuracy for questions rated below 4.0. This finding validates the role of expert annotation in benchmark curation, especially for spotting questions where the intended correct answer may itself be debatable.

B. Augmentation

The `+think` and `+tool` results have direct practical implications. Reasoning augmentation helps free-response consistently but can destabilize MCQ performance near the ceiling. The MCQ drop under +think appears to come from overthinking. One possible explanation is that extended reasoning chains lead to models incorrectly revising correct answers. Tool augmentation demands a minimum model capacity (≈8B parameters), below that threshold, models produce malformed tool invocations or get stuck in repeated search loops that exceed the iteration limit without adding useful information. From a practitioner's standpoint, a Mistral 3B base model (1.3 s/question, 3.54 FR overall) may beat a Llama3.2 3B +think variant (4.9 s/question, 3.01 FR overall) when balancing quality against throughput. For the highest free-response quality, Mistral 14B base (2.4 s, 4.33 FR overall) offers the best local-model trade-off, approaching GPT-5-Mini (4.52) at a fraction of the latency.

C. Implications and Limitations

Models are reliable for MCQ-style knowledge tasks: base models with 3B+ parameters answer recall, analysis, and design questions correctly over 90% of the time. Free-response evaluation reveals a capability gap that MCQ accuracy alone does not reflect [24]. These results address three distinct audiences.

For practitioners, model selection should hinge on generative evaluation rather than multiple-choice accuracy [8]. The conviction metric supplies an added signal, suggestions produced with unanimous conviction (89.5% accuracy) can be trusted more than split-majority responses (55.0%).

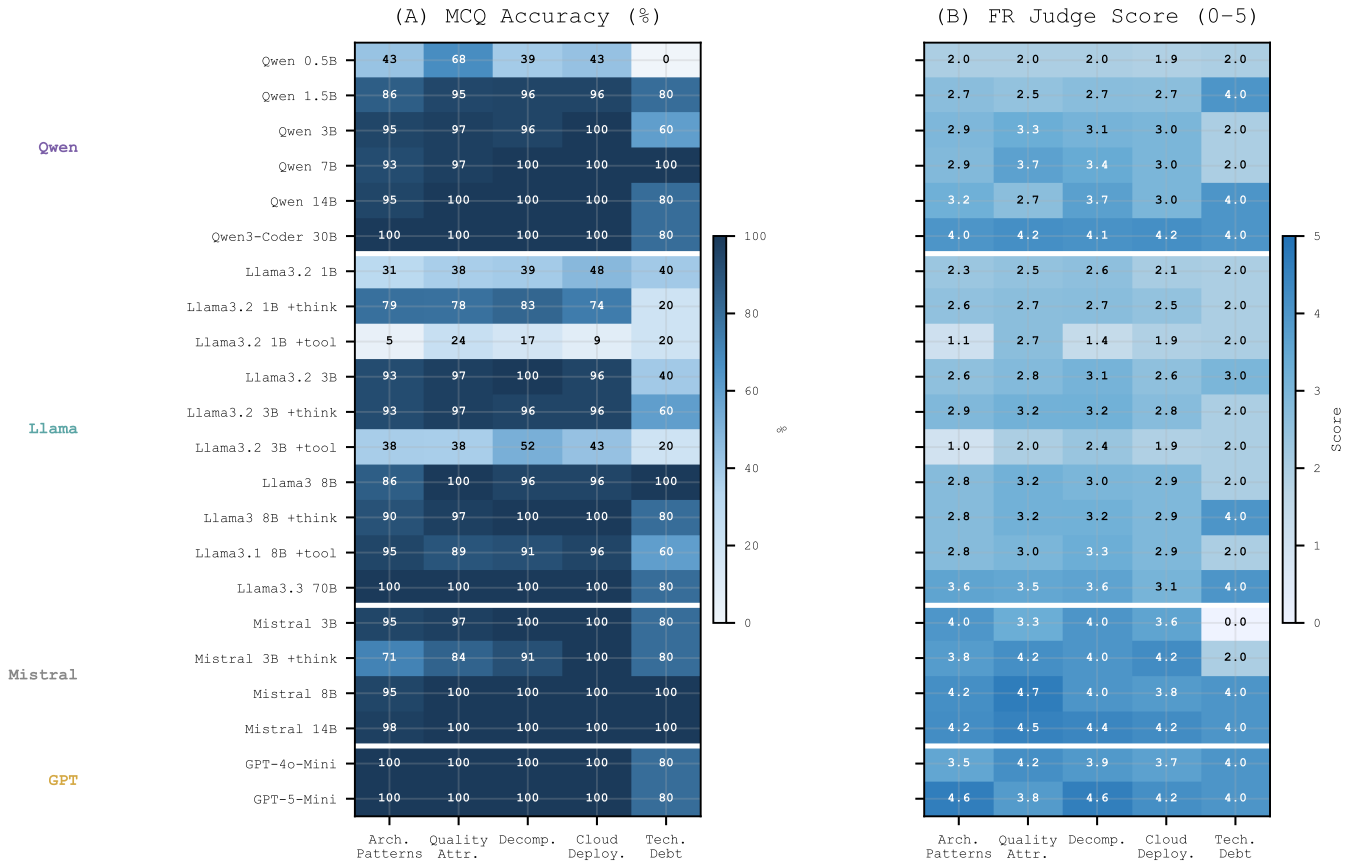

Fig. 7. Per-topic performance across all 22 configurations. (A) MCQ accuracy (%) aggregated across recall, analyze, and design levels. (B) free-response judge scores (0–5) aggregated across analyze, design, and implement levels. Color intensity indicates performance; white separators mark family boundaries.

For educators, the cognitive-level breakdown maps straight to Bloom's taxonomy [13], helping to point out which competencies can be handed off to LLMs. Recall and analysis tasks are well within reach of even small models, while design and implementation require larger models or human oversight.

For tool builders, free-response quality suggests LLMs work as effective first-draft generators for architectural artifacts [12], as long as a human architect reviews the output. The wide variance in implement-level scores (1.36–4.54) means implementation assistance should come with confidence-aware guardrails.

The study reveals six key limitations.
1) Scope: CAKE targets cloud-native architecture and results may not transfer to classical patterns.
2) MCQ distractor quality: Expert review detected elaboration bias in some distractors. Option shuffling mitigates positional bias and CAKE-Core addresses this with strict quality filtering.
3) Excluded implement MCQs: 12 implement-level MCQ items were dropped due to a formatting defect, since implementation knowledge is better assessed through free-response questions, the final MCQ set comprised 130 items.
4) Judge model: Free-response relies on a single primary judge (DeepSeek-R1:32B), partially mitigated by validation with Gemini 2.5 Pro.
5) Inter-rater reliability: Ordinal Krippendorff's α values are near zero, a known artifact when distributions are heavily skewed [16]. While 91.3% within-one-point agreement supports practical consensus, future iterations should consider wider scales.
6) Correct answer patterns: Questions tend to have the longest option as the correct answer, which could allow heuristic exploitation; future iterations should control distractor length.

VI. Conclusion

CAKE is the first benchmark built for cloud-native software architecture knowledge across cognitive levels. Our evaluation of 22 model configurations surfaces four findings. First, MCQ accuracy reaches near-ceiling for models with 3B+ parameters. Second, free-response scores scale steadily across all three cognitive levels, including implement (best: 4.54/5). Third, the two evaluation formats expose complementary facets of knowledge, as MCQ saturates while free-response keeps differentiating models across the full parameter range. Finally augmentation effects depend on size, with reasoning enhancement lifting free-response quality while tool augmentation leads to degradation below ≈8B parameters.

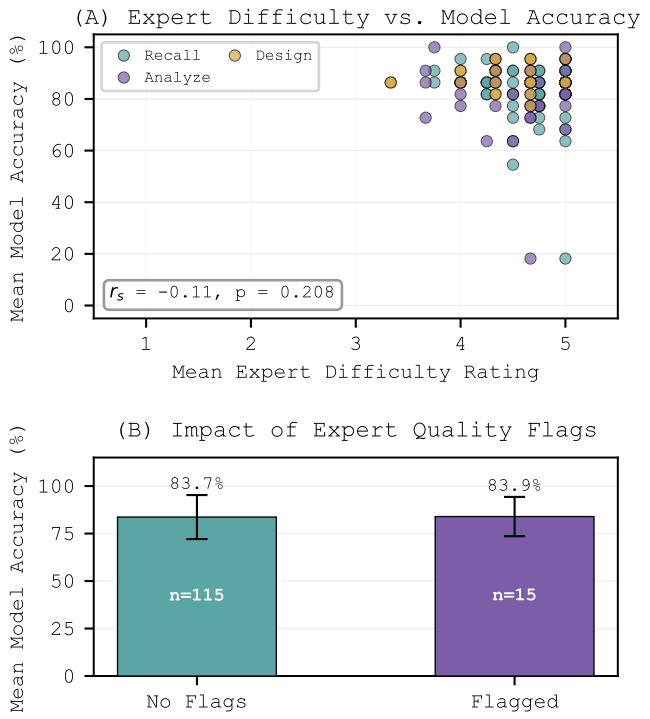

Fig. 8. Expert–model alignment analysis. (A) Expert difficulty ratings show no significant correlation with mean model accuracy across 22 configurations. (B) Questions flagged for ambiguity or typos show no accuracy difference from unflagged questions.

These findings extend beyond software architecture. The gap between MCQ and free-response results indicates that benchmarks relying solely on multiple-choice questions may overestimate model capabilities, particularly in domains demanding procedural and design knowledge. Moreover, the conviction metric demonstrates that multi-run evaluation provides actionable confidence signals with no additional cost beyond increased inference time.

Future work will address the coverage gap by incorporating correctly formatted implementation-level MCQs, expand the scope beyond cloud-native architectures to include classical software architecture patterns, investigate optimal augmentation strategies across different model sizes, and extend the benchmark to multilingual settings.

Data Availability

The CAKE dataset is publicly available at https://github.com/timadam03/CAKE-benchmark.

References

[1] C. E. Jimenez, J. Yang, A. Wettig, S. Yao, K. Pei, O. Press, et al., "SWE-bench: Can language models resolve real-world GitHub issues?", in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024.

[2] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, J. Kaplan, et al., "Evaluating large language models trained on code," *arXiv preprint arXiv:2107.03374*, 2021.

[3] H. Han, J. Kim, J. Yoo, Y. Lee, and S.-w. Hwang, "ArchCode: Incorporating software requirements in code generation with large language models," in *Proc. Annu. Meet. Assoc. Comput. Linguist. (ACL)*, 2024.

[4] A. Gu, B. Roziere, H. Leather, A. Solar-Lezama, G. Synnaeve, and S. I. Wang, "CRUXEval: A benchmark for code reasoning, understanding, and execution," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2024.

[5] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, et al., "Measuring massive multitask language understanding," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.

[6] A. Srivastava, A. Rastogi, A. Rao, A. A. M. Shoeb, A. Abid, et al., "Beyond the imitation game: Quantifying and extrapolating the capabilities of language models," *Trans. Mach. Learn. Res.*, 2023.

[7] D. Rein, B. L. Hou, A. C. Stickland, J. Petty, R. Y. Pang, J. Dirani, et al., "GPQA: A graduate-level Google-proof Q&A benchmark," *arXiv preprint arXiv:2311.12022*, 2023.

[8] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, et al., "Judging LLM-as-a-judge with MT-Bench and Chatbot Arena," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2023.

[9] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*, 4th ed. Boston, MA, USA: Addison-Wesley, 2021.

[10] M. Richards and N. Ford, *Fundamentals of Software Architecture*. Sebastopol, CA, USA: O'Reilly Media, 2020.

[11] R. Caceffo, S. Wolfman, K. S. Booth, and R. Azevedo, "Developing a computer science concept inventory for introductory programming," in *Proc. ACM Tech. Symp. Comput. Sci. Educ. (SIGCSE)*, 2016, pp. 364–369.

[12] M. Esposito, X. Li, S. Moreschini, N. Ahmad, T. Cerny, K. Vaidhyanathan, et al., "Generative AI for software architecture: Applications, challenges, and future directions," *arXiv preprint arXiv:2503.13310*, 2025.

[13] L. W. Anderson and D. R. Krathwohl, Eds., *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. New York, NY, USA: Longman, 2001.

[14] E. Thompson, A. Luxton-Reilly, J. L. Whalley, M. Hu, and P. Robbins, "Bloom's taxonomy for CS assessment," in *Proc. Australas. Comput. Educ. Conf. (ACE)*, 2008, pp. 155–161.

[15] U. Fuller, C. G. Johnson, T. Ahoniemi, D. Cukierman, I. Hernán-Losada, J. Jackova, et al., "Developing a computer science-specific learning taxonomy," *ACM SIGCSE Bull.*, vol. 39, no. 4, pp. 152–170, 2007.

[16] K. Krippendorff, *Content Analysis: An Introduction to Its Methodology*, 4th ed. Thousand Oaks, CA, USA: Sage, 2018.

[17] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, et al., "Scaling laws for neural language models," *arXiv preprint arXiv:2001.08361*, 2020.

[18] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, et al., "Qwen2.5 technical report," *arXiv preprint arXiv:2412.15115*, 2024.

[19] S. Prakash, A. Cheng, J. Yik, A. Tschand, R. Ghosal, I. Uchendu, et al., "QuArch: A question-answering dataset for AI agents in computer architecture," *IEEE Comput. Archit. Lett.*, vol. 24, no. 1, pp. 105–108, 2025.

[20] A. S. Thakur, K. Choudhary, V. S. Ramayapally, S. Vaidyanathan, and D. Hupkes, "Judging the judges: Evaluating alignment and vulnerabilities in LLMs-as-judges," in *Proc. GEM Workshop*, in conjunction with *ACL*, 2025.

[21] Z. Fei, X. Shen, D. Zhu, F. Zhou, Z. Han, A. Huang, et al., "LawBench: Benchmarking legal knowledge of large language models," in *Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, 2024.

[22] S. Chen, L. H. Khiem, A. Szymanski, R. Metoyer, T. Hua, and N. V. Chawla, "Automated benchmark generation from domain guidelines informed by Bloom's taxonomy," *arXiv preprint arXiv:2601.20253*, 2026.

[23] A. C. Doris, D. Grandi, R. Tomich, M. F. Alam, M. Ataei, H. Cheong, et al., "DesignQA: A multimodal benchmark for evaluating large language models' understanding of engineering documentation," *J. Comput. Inf. Sci. Eng.*, vol. 25, no. 2, art. no. 021009, 2025.

[24] L. Schmid, T. Hey, M. Armbruster, S. Corallo, D. Fuchß, J. Keim, et al., "Software architecture meets LLMs: A systematic literature review," *arXiv preprint arXiv:2505.16697*, 2025.

[25] Y. Wang, X. Ma, G. Zhang, Y. Ni, A. Chandra, S. Guo, et al., "MMLU-Pro: A more robust and challenging multi-task language understanding benchmark," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2024.

[26] X. Fu and W. Liu, "How reliable is multilingual LLM-as-a-judge?" in *Findings of the Association for Computational Linguistics (EMNLP)*, 2025.